



Intégrer son logiciel dans le catalogue ESR

Guide de bonnes pratiques pour bien remplir une notice logicielle sur HAL

Version : mars 2026

Ce guide s'adresse aux auteurs et gestionnaires de dépôts qui souhaitent référencer un logiciel de recherche sur hal.science de façon à ce que la notice soit correctement moissonnée et affichée dans le catalogue des logiciels de l'ESR.

■ Sommaire

1	Contexte et enjeux	3
2	Préparer le dépôt avant de remplir la notice	4
2.1	Informations nécessaires à inclure dans le code repository	4
2.2	Archiver dans Software Heritage (SWH)	5
3	Remplir la notice HAL : champ par champ	6
3.1	Métadonnées obligatoires	6
3.2	Métadonnées fortement recommandées	7
3.3	Métadonnées complémentaires	7
4	Conseils de rédaction	8
4.1	Le champ Nom du logiciel (Titre)	8
4.2	Le champ Description	8
4.3	Les mots-clés	8
4.4	La licence	9
5	Checklist avant de valider le dépôt	10
6	Après le dépôt : validation technique et mise à jour	11
6.1	La phase de validation technique	11
6.2	Délai d'apparition dans le catalogue ESR	11
6.3	Mettre à jour une notice existante	11
7	Ressources utiles	12

■ 1 Contexte et enjeux

Le catalogue des logiciels de l'Enseignement Supérieur et de la Recherche (ESR) est une initiative du Ministère de l'Enseignement Supérieur et de la Recherche (MESR), portée par le [Comité pour la Science Ouverte](#). Son objectif est de rendre visible les logiciels produits dans les laboratoires de recherche académiques français, de faciliter leur recensement et d'établir des liens entre publications, données et codes sources.

Le catalogue, dans sa version beta actuelle, est alimenté par moissonnage automatique de HAL qui est réalisé toutes les nuits. Il récupère toutes les notices de type « Logiciel », disposant d'un SWHID (identifiant Software Heritage). La qualité de ce qui apparaît dans le catalogue dépend donc directement de la qualité de la notice HAL. A terme, une étape de modération permettra de s'assurer de la qualité des métadonnées.

i Principe fondamental

Ce que vous saisissez dans HAL est ce qui sera visible dans le catalogue ESR. Des métadonnées incomplètes ou mal rédigées impacteront la lisibilité de la fiche du catalogue associée et la visibilité du logiciel. A terme, ces logiciels ne seront plus intégrés dans le catalogue.

■ 2 Préparer le dépôt avant de remplir la notice

■ 2.1 Informations nécessaires à inclure dans le code repository

Avant de créer la notice HAL, assurez vous que votre dépôt de code (sur une forge logicielle idéalement souveraine) contient toutes les informations nécessaires pour son ouverture. C'est la même chose pour une notice réalisée à partir d'une archive du code (zip par exemple), mais cette façon de faire est à déconseiller dans la mesure du possible.

Informations	Importance	Où l'intégrer ?
Licence	Obligatoire	La licence associée au logiciel doit se trouver sous la forme d'un fichier texte (qui contient tous les termes du contrat de licence) à la racine du dépôt. Ce fichier se nomme généralement LICENSE ou COPYING .
Auteurs/ autrices & copyright	Obligatoire	L'indication des auteurs du code permet l'attribution des droits et l'identification des propriétaires des droits patrimoniaux c'est-à-dire les employeurs des auteurs qui disposent du copyright. Elle peut se faire dans un fichier spécifique souvent nommé AUTHORS ou dans l'entête du texte de la licence, ou dans le fichier README.md .
Description	Fortement recommandé	En général, la description du logiciel se trouve dans le fichier README.md qui est la première chose qu'on voit sur le dépôt.
Installation	Recommandé	Les informations concernant l'installation se trouvent généralement dans le fichier README.md ou dans un fichier INSTALL à part.
Usage	Recommandé	Le fichier README.md comprend en général une partie expliquant la façon d'utiliser le logiciel.

Les informations obligatoires seront vérifiées lors de la validation technique de HAL (voir section 6.1).

Outre les fichiers mentionnés ci-dessus, il peut être intéressant de créer à la racine du dépôt du code un fichier **codemeta.json**. Ce fichier structuré au format **CodeMeta** permet de regrouper les métadonnées descriptives du logiciel qui pourront être utilisées pour pré-remplir la notice HAL.

✓ Générer un fichier **codemeta.json**

Le fichier **codemeta.json** se génère facilement via différents outils :

- **CodeMeta generator**.
- **Auto CodeMeta generator**, qui permet d'extraire d'un dépôt un certain nombre d'informations automatiquement.
- **Codemetapy**, outil python en ligne de commande pour extraire les métadonnées

d'un package Python, Java, NodeJS... pour créer le fichier `codemeta.json` associé.

- **codemeta**, outil qui génère automatiquement un fichier `codemeta.json` à partir d'un package R.

■ 2.2 Archiver dans Software Heritage (SWH)

Software Heritage archive automatiquement les dépôts publics des principales forges. Vous pouvez vérifier et forcer l'archivage via la fonction [Save code now](#).

Une fois archivé, copiez le **SoftWare Hash Identifier (SWHID, avec les informations de contexte)** de la version que vous souhaitez référencer. Il se présente sous la forme :

```
swh:1:dir:xxxxxxxxxxxxxxxxxxxxxxxxxxxx;origin=xxxxxxxx;visit=xxxxxxxxxxxxxxxx;
```

Ce SWHID, copié / collé dans la notice HAL, permettra de déclencher l'extraction automatique des métadonnées depuis le fichier `codemeta.json` si celui-ci existe. Si ce n'est pas le cas, il assurera de toutes façons le lien entre la notice HAL et l'archive du code sur Software Heritage.

Il est important d'utiliser la fonctionnalité « Copy identifier » dans le tiroir latéral « Permalinks » sur l'archive du code sur Software Heritage, pour coller le SWHID dans la notice HAL.

⚠ Important

Dans le cas d'une notice réalisée à partir d'un fichier zip du code, il est nécessaire d'**accepter** l'archivage sur Software Heritage lors du dépôt HAL pour qu'un SWHID soit inclus dans la notice. Sans cette étape, le code ne sera pas archivé de façon pérenne. Et n'apparaîtra pas non plus dans le catalogue.

📘 Pour plus d'informations sur l'archivage

Vous pouvez vous reporter à la [documentation dédiée](#).

■ 3 Remplir la notice HAL : champ par champ

Connectez-vous à hal.science, cliquez sur « Déposer », puis choisissez le type de document « Logiciel ».

- Si vous avez un fichier *codemeta.json* dans le dépôt de votre code, collez votre SWHID contextualisé dans la section « Extraction automatique » pour pré-remplir les champs. Vérifiez ensuite chaque champ.
- Si vous n'avez pas de fichier *codemeta.json*, indiquez le SWHID dans le champ « identifiants » de la section « informations principales ».

■ 3.1 Métadonnées obligatoires

Ces champs sont indispensables pour que le dépôt soit validé et le logiciel correctement affiché dans le catalogue.

Information	Champ HAL	Description / Recommandation
Auteurs/ autrices & affiliations	Auteurs et affiliations	Ajoutez tous les auteurs via leur IdHAL si possible. L'affiliation est essentielle pour informer du copyright. Elle est également cruciale pour la vue « structure » du catalogue.
Nom du logiciel	Nom	Saisissez le nom officiel avec la casse exacte. Évitez les abréviations ambiguës et utilisez le nom court. C'est le nom affiché en tête de fiche dans le catalogue.
Domaine thématique	Domaine	Sélectionnez le ou les domaines HAL correspondants. Détermine dans quels domaines le logiciel est indexé. Permet d'utiliser certains filtres dans le catalogue.
Description	Description	Rédigez un paragraphe clair et synthétique au format texte simple expliquant à quoi sert le logiciel, pour qui, et quels problèmes il résout. Doit être cohérent avec le README. Évitez le jargon trop technique sans contexte. Cette description est le principal vecteur de découvrabilité dans le catalogue.
Licence	Licence	Choisissez la licence en utilisant les identifiants du référentiel spdx .

⚠ Affiliation

Les entrées (laboratoires, établissements) doivent être de couleur **verte** (valide). Elles peuvent être **jaune** (structure qui n'existe plus depuis peu) en fonction de la date de production du code.

■ 3.2 Métadonnées fortement recommandées

Ces champs ne sont pas obligatoires mais sont directement moissonnés et affichés dans le catalogue. Leur absence appauvrit significativement la fiche.

Information	Champ HAL	Description / Recommandation
Mots-clés	Mots-clés	Ajoutez quelques mots-clés pertinents, en français et/ou anglais. Utilisez des termes issus du domaine : noms de méthodes, algorithmes, champs disciplinaires. Pensez aux termes que de futurs utilisateurs emploieraient, autant qu'aux termes utilisés par les personnes intéressées par la méthodologie du logiciel en elle-même.
URL du dépôt du code	Code Repository	Métadonnée non obligatoire mais très fortement recommandée . Lien vers le dépôt du code sur une forge. Vérifiez que le lien n'est pas brisé. Ce lien servira à enrichir les informations du catalogue.
Publications et jeux de données associés	Ressources associées	Liez les publications qui décrivent ou utilisent le logiciel, ainsi que les jeux de données déposés dans un entrepôt adapté. Renforce les liens publications/données/logiciels dans le catalogue.
Site du logiciel	Voir aussi	Lorsque le logiciel dispose de son propre site (différent du dépôt du code).

■ 3.3 Métadonnées complémentaires

Ces champs enrichissent la fiche mais ne sont pas essentiels pour le catalogue.

Information	Champ HAL	Description / Recommandation
Langage(s) de programmation	Langage de programmation	Indiquez le ou les langages (Python, C++, Julia, R...). Séparez les par des virgules si il y en a plusieurs. Utile pour le filtrage dans le catalogue.
Financements (ANR, Europe...)	Financements	Liez les projets ANR ou européens via les champs dédiés. Les projets ANR sont récupérables depuis un référentiel intégré à HAL.

■ 4 Conseils de rédaction

■ 4.1 Le champ Nom du logiciel (Titre)

- Formatez le nom selon la forme **Acronyme** ou **Acronyme : nom développé**
- Evitez les caractères spéciaux de type crochets ou parenthèses.
- Evitez les titres trop longs.
- Utilisez le nom officiel tel qu'il apparaît dans le README et dans les publications.
- Ne mettez pas les numéros de version dans le titre (ils vont dans le champ Version).
- Respectez la casse : le nom tel qu'il doit être cité.

■ 4.2 Le champ Description

Répondez de façon synthétique en utilisant un format texte simple (sans balisage) aux questions suivantes :

- Qu'est-ce que ce logiciel fait ?
- Pour quel domaine scientifique ou applicatif ?
- Quelles sont ses fonctionnalités principales ?
- Quelles sont ses particularités par rapport à l'état de l'art ?
- Qui sont les utilisateurs cibles (chercheurs, ingénieurs, grand public...) ?

✓ Exemple de bonne description

« MonLogiciel est un outil de simulation numérique pour la modélisation du transfert radiatif dans les atmosphères planétaires. Il implémente les équations de transfert en géométrie sphérique et supporte les calculs parallèles (MPI). Développé pour les besoins de la communauté planétologie, il a été utilisé dans plus de 15 publications depuis 2018. »

⚠ Exemples de descriptions insuffisantes à éviter

- « Code Python pour la physique » : trop vague.
- « Voir le README » : inutilisable dans le catalogue.
- « Logiciel développé au laboratoire XYZ » : n'explique pas la fonction.

■ 4.3 Les mots-clés

- Utilisez des termes qui seront pertinents dans le cadre d'une recherche textuelle.
- Combinez des termes généraux (domaine disciplinaire) et des termes spécifiques (méthodes, algorithmes).

- Ajoutez des termes en anglais pour maximiser la visibilité internationale.
- Exemple pour un logiciel de simulation moléculaire : « *dynamique moléculaire* », « *molecular dynamics* », « *champ de forces* », « *force field* », « *bioinformatique* ».

■ 4.4 La licence

Quelques points importants :

- Utilisez un format standard pour l'intitulé en utilisant les [identifiants spdx](#).
- Ne diffusez pas votre logiciel sans licence. Si vous avez des questions à ce sujet, rapprochez vous du service valorisation de votre établissement.
- La licence déclarée dans HAL doit être identique à celle du fichier LICENSE ou COPYING dans le repository.

■ 5 Checklist avant de valider le dépôt

Avant de soumettre le dépôt à la validation technique, vérifiez chaque point :

Point à vérifier	
☐	Le nom du logiciel est court et sans numéro de version.
☐	La description est rédigée en prose, claire et informative.
☐	Plusieurs mots-clés sont renseignés (français et/ou anglais).
☐	La licence est bien formatée (liste spdx) et correspond au fichier LICENSE ou COPYING du repository.
☐	Les auteurs et leurs affiliations sont correctes.
☐	Le lien vers le code repository est fonctionnel.
☐	Le repository (ou le fichier archive déposé) comprend au moins un README et un fichier licence (LICENSE / COPYING) et les auteurs et le copyright y sont bien mentionnés.
☐	Le SWHID est bien renseigné.
☐	Les publications et jeux de données associés sont liés via « Ressources associées ».

■ 6 Après le dépôt : validation technique et mise à jour

■ 6.1 La phase de validation technique

Tout dépôt de notice logicielle dans HAL est soumis à une validation technique réalisée par des professionnels de l'information scientifique avant mise en ligne. Le valideur vérifie la cohérence entre les métadonnées et le code déposé, notamment : la présence des fichiers README, et LICENSE ou COPYING, l'indication des auteurs et de leurs affiliations (copyright), la cohérence du titre et de la description avec le contenu du code.

Si des informations manquent ou sont incohérentes, le valideur peut contacter l'auteur de la notice pour corrections.

■ 6.2 Délai d'apparition dans le catalogue ESR

Le catalogue moissonne HAL toutes les nuits. Après validation de votre dépôt sur HAL, la fiche apparaîtra très rapidement dans le catalogue. Les mises à jour de métadonnées sur HAL sont répercutées lors du prochain cycle de moissonnage.

■ 6.3 Mettre à jour une notice existante

Si votre logiciel évolue (nouvelle version, changement de licence, nouveaux auteurs...), déposez une nouvelle version dans HAL via **Mon espace / Mes dépôts / Déposer une nouvelle version**. Les métadonnées peuvent alors être modifiées. N'oubliez pas de mettre à jour le SWHID associé à cette nouvelle version.

■ 7 Ressources utiles

Documentation dépôt logiciel HAL

<https://doc.hal.science/deposer/deposer-le-code-source/>

Software Heritage - Save Code Now

<https://archive.softwareheritage.org/save/>

CodeMeta Generator

<https://codemeta.github.io/codemeta-generator/>

Catalogue des logiciels de l'ESR

<https://logiciels.catalogue-esr.fr/>

Guide dépôt logiciel HAL (PDF Inria)

<https://inria.hal.science/hal-01872189>

Contact catalogue ESR

contact-logiciels-catalogue-esr@groupes.renater.fr